Spotify Song Recommender

Rishi Kamtam

Problem Statement

Which one should we use

Do higher levels of a song's characteristics correlate with the amount of time you have spent listening to a song?

We can create a classifier that can predict whether somebody would like a new song depending on the characteristics?

01 | The Problem

Why this relationship is important

- There is an abundance of music suggestions
  - Friend recommendation, new albums, family’s favorites
- Time is scare
  - Listening to a song takes on average 3 minutes and 30 seconds
  - Ex. 5 album’s from popular artists were dropped at midnight on 04/05/2024
    - Total listen time: 4 hours and 18 minutes
- The insight derived from correlation makes recommendation easier

Goals of the Project

- Gather and normalize data (song features)
- Do multiple linear regression on time played and each feature
- Create a KNN classifier to see if a recommended song would be enjoyed by listener

02| Gathering the DATA

Introduction to the Data and where it was sourced from.

- Acquired through spotify
- Obtained in two manners
  - **Json of listening history (from spotify)**
  - Spotipy Api for features

Feature Data

MsPlayed

Json from spotify data

Name	Duration played	Spotify ID
Name of song	Total ms played	Id spotify assigns to song

The Features

What does spotify mean by features

- Spotify calculates audio features for each track
  - features are given a numeric value on a scale
- Access features through Spotify's api

What are the features

Acousticness, Danceability, Energy, Instrumentalness, Liveness, Loudness, Speechiness, Valence, Tempo

Library's

- Spotipy
- Requests
- Pandas
- Numpy
- Seaborn
- SKlearn
- OS

Regression

Our Correlation Findings

For the Whole Dataset

R^2 val: 0.01352

Features:

Danceability: **-0.0218**
Energy: **-0.0255**
Key: **-0.0131**
Loudness: **0.0215**
Mode: **0.0208**
Speechiness: **-0.079**
Acousticness: **0.0137**
Instrumentalness: **-0.0497**
Liveness: **0.0118**
Valence: **-0.0302**
Tempo: **-0.013**

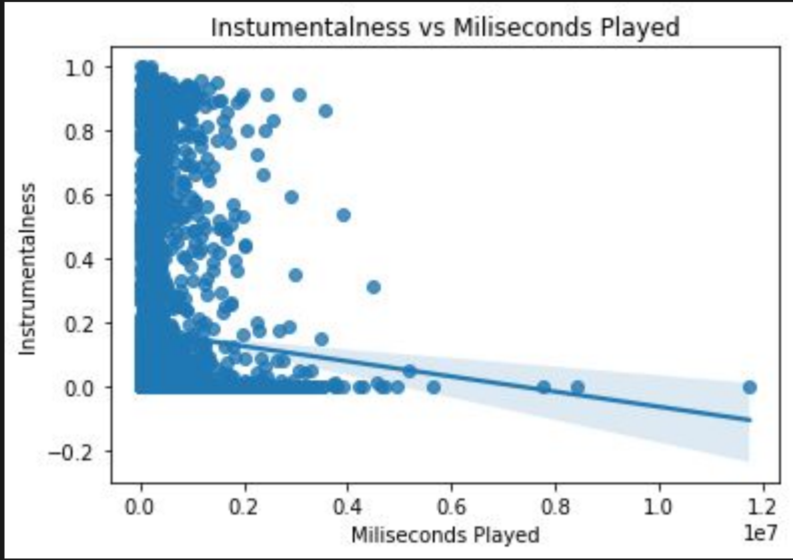
For the songs with >10 mins played

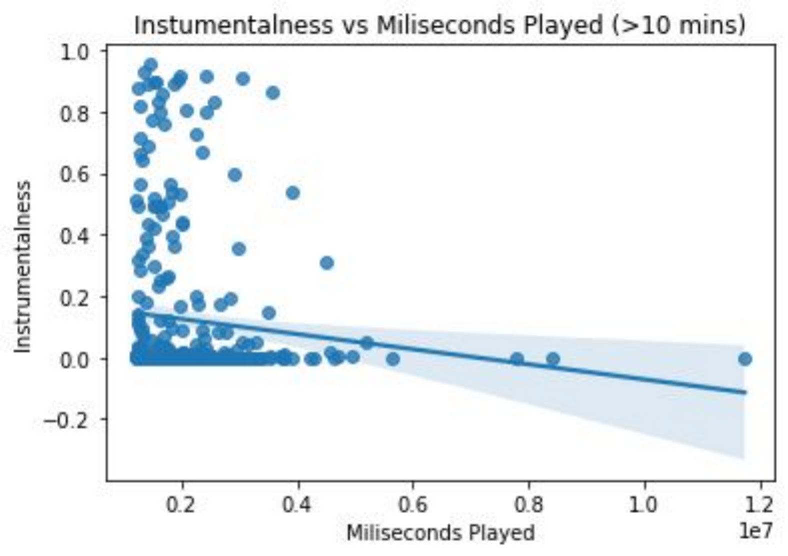
R^2 val: 0.03847

Features:

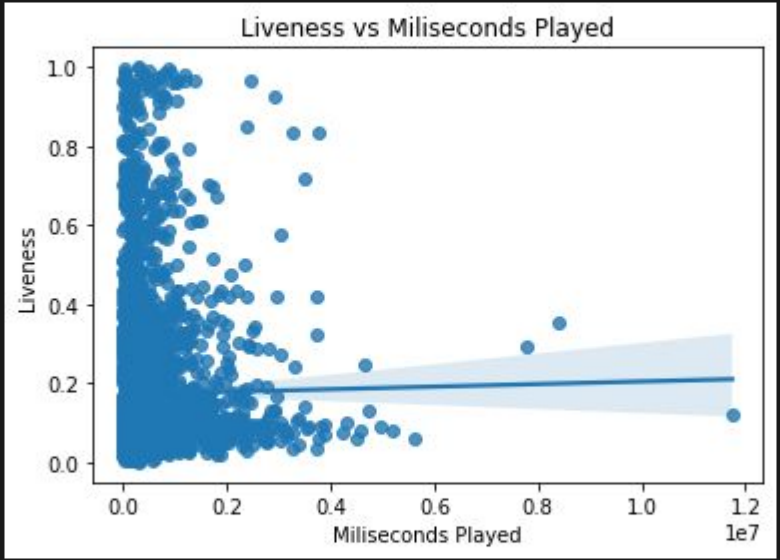
Danceability: **-0.0413**
Energy: **-0.0339**
Key: **-0.0055**
Loudness: **0.075**
Mode: **-0.0013**
Speechiness: **0.0395**
Acousticness: **0.012**
Instrumentalness: **-0.1043**
Liveness: **0.033**
Valence: **-0.035**
Tempo: **-0.0808**

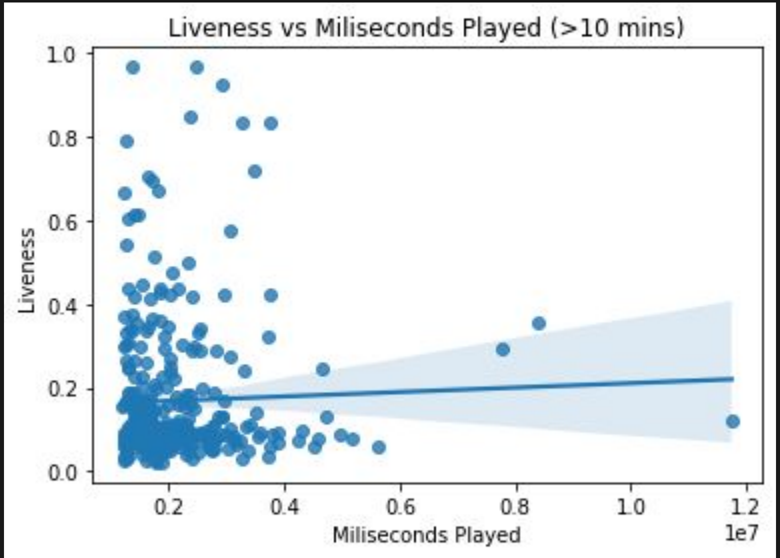
Visualization - Instrumentalness



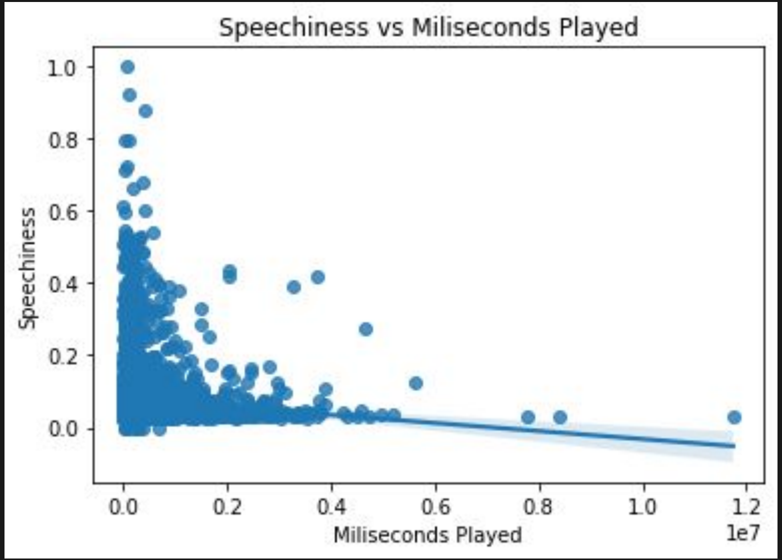


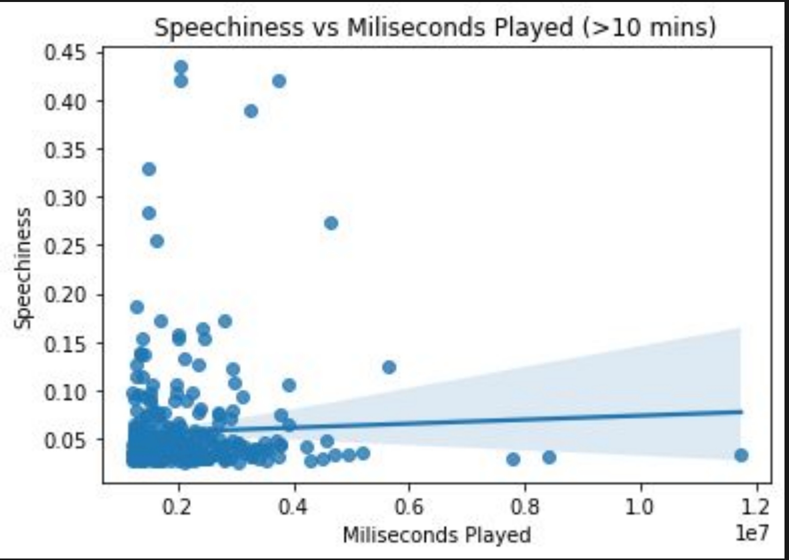
Visualization - Liveness





Visualization - Speechiness





K-Nearest Neighbors Classifier

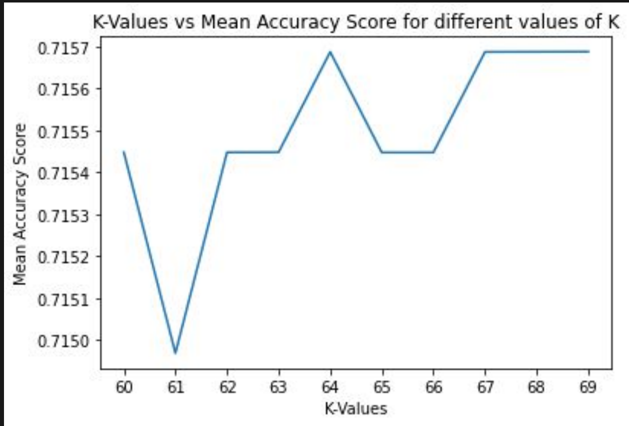
Finding Optimal K-Value

Optimal K Value:

- Research findings that optimal K value for a KNN classifier located around the square root of the amount of data
- We had 4169 pieces of data-> Square root is roughly 65

Range we used was between 60 and 70, within 5 of square root

- To further find the optimal K value between 60-70, decided to use mean accuracy
- Utilized K-Fold and Cross validation to get a mean accuracy score
- Accuracy = (Correct Predictions / Total Instances)
- K value between 60 and 70 with the highest mean accuracy
    - 69 (71.57%)



KNN Classifier Results

Features:

- Normalized versions of: Acousticness, Danceability, Energy, Instrumentalness, Liveness, Loudness, Speechiness, Valence, Tempo

Labels: Yes or No

- Yes label given to songs above the mean of Ms Played
- No labels given to songs below the mean of Ms Played
- Mean of Ms Played was 433104.35 (Roughly 7.2 minutes)

Scores:

Accuracy: 73.896%

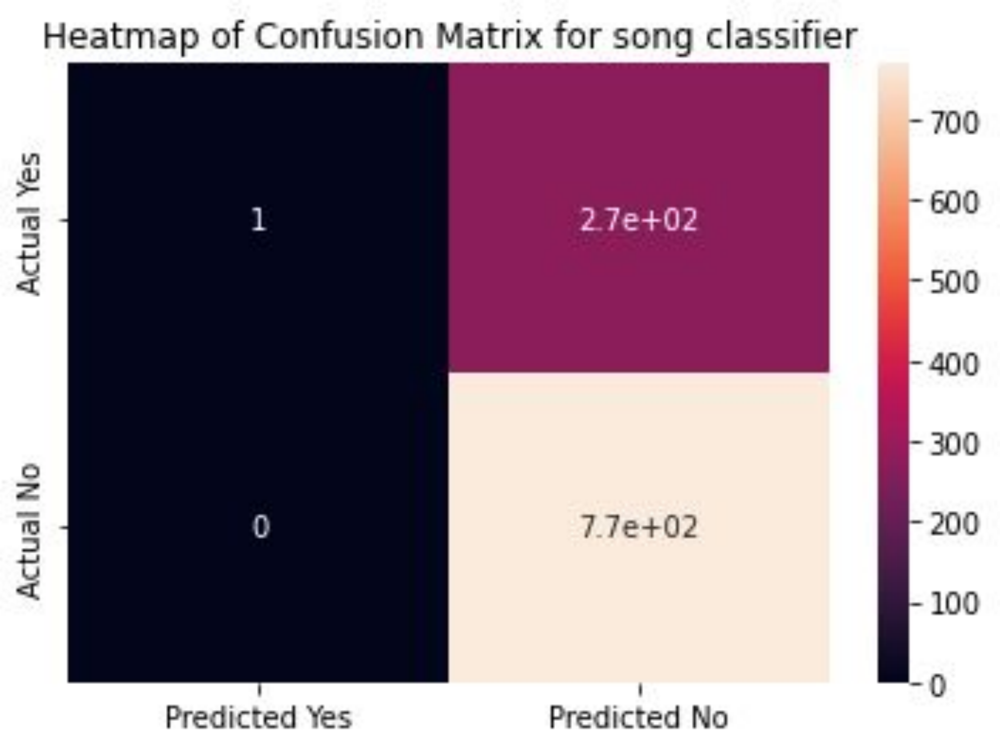
Precision: 100%

Recall: 0.366%

F1 Score: 0.73%; (Harmony between Precision and Recall) ;

- Classifier is not effectively identifying and correctly classifying positive instances

Heatmap of Confusion Matrix



- Supports metrics
- **Great** at predicting **No** label (770 No, Predicted No)
- **Poor** at predicting **Yes** label (270 Yes, Predicted No)

Predicting Specific Songs

- Next step was using classifier to predict labels for recommended songs
- Tested 10 songs
    - 7 newly recommended songs and 3 already existing songs
    - 2 of the 3 already existing songs were already given the “Yes” label
    - **All 10** songs were predicted “**No**” by classifier

Next Steps

How can we improve?

- Changing scope → playlist size
- Add weighting by artist for each individual
    - Accounts for people liking certain artists more
- Analyze by different variables
    - Look at songs by genre, age, etc.

Conclusion

Data:

- Gathered through Spotify's API and JSON of Jeff's listening history

Correlation/Regression:

- Weak correlation findings between features and time played

Classifier:

- Weak correlation a contributor to poor classifier metrics